

CS247: ADVANCED DATA MINING


06: Text Data: Word Embedding

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

January 31, 2024

Text Data: Word Embedding

- Introduction to Word Representation 
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary

Why Word Representation?

- Finding Synonyms: words that have the same meaning
 - E.g., movie and film
- Finding polysemy: words with multiple meanings
 - E.g., light
- Document representation
 - E.g., aggregation of all the word representation

How to Represent a Word?

- Challenge
 - Discrete structure
- Simple representation
 - One-hot representation: a vector with one 1 and a lot of zeroes
 - E.g., Motel =

[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

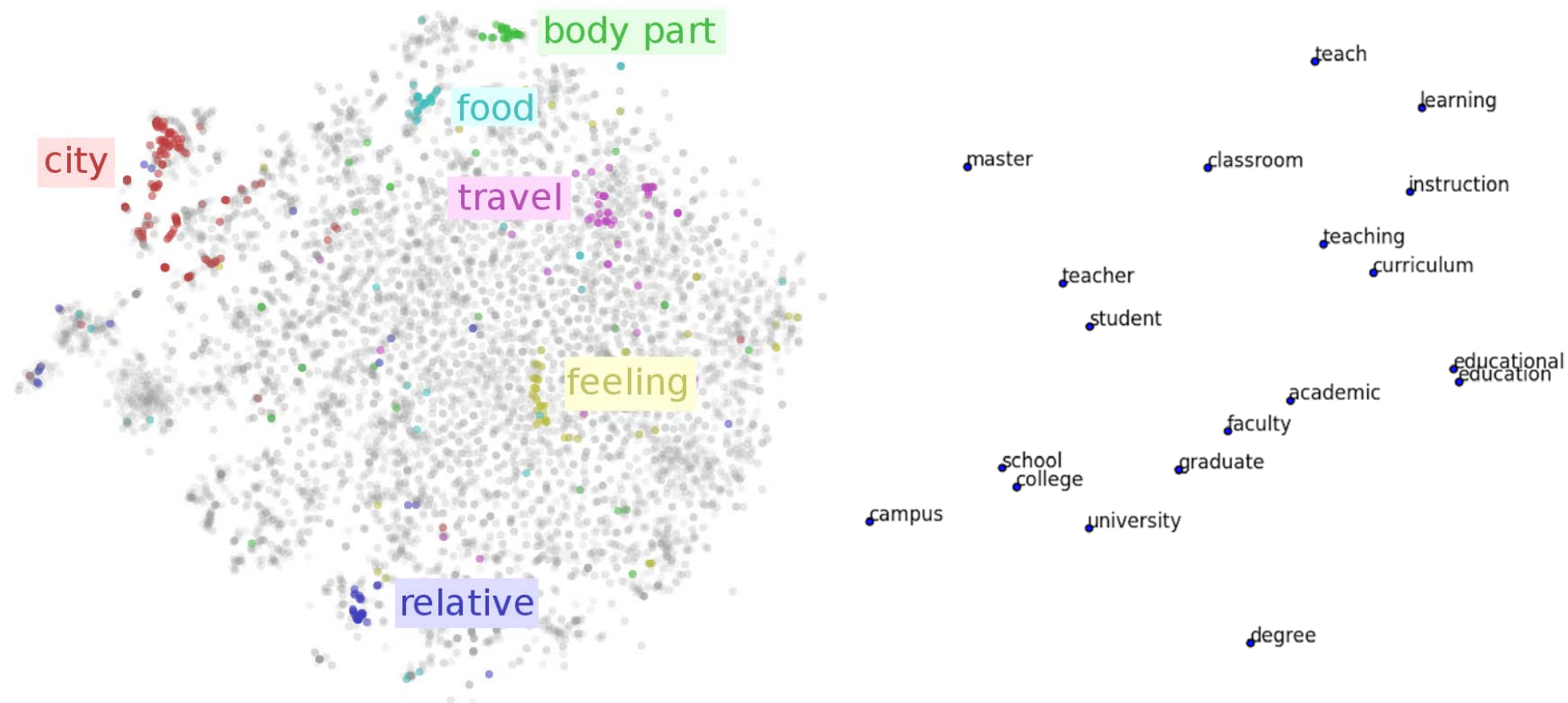
Problem of One-Hot Representation

- High dimensionality
 - E.g., for Google news, 13M words
- Sparse
 - Only 1 non-zero value
- Shallow representation
 - E.g.,

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Word Embedding

- Low dimensional vector representation of every word
 - E.g., motel = [1.3, -1.4] and hotel = [1.2, -1.5]



How to Learn Such Embeddings?

- Using context information!

...he curtains open and the sun is shining in on the barely...
...ars and the cold , close to the ground . And neither of the w...
...rough the night with the moon shining so brightly , it...
...made in the light of the sun . It all boils down , wr...
...surely under a crescent moon , thrilled by ice-white...
...sun , the seasons of the year ? Home , alone , Jay pla...
...m is dazzling snow , the moon has risen full and cold...
...un and the temple of the sun , driving out of the hug...
...in the dark and now the moon rises , full and amber a...
...bird on the shape of the moon over the trees in front...

How to Learn Such Embeddings?

- Using context information!

...he curtains open and the moon shining in on the barely...
...ars and the cold , close moon " . And neither of the w...
...rough the night with the moon shining so brightly , it...
...made in the light of the moon . It all boils down , wr...
...surely under a crescent moon , thrilled by ice-white...
...sun , the seasons of the moon ? Home , alone , Jay pla...
...m is dazzling snow , the moon has risen full and cold...
...un and the temple of the moon , driving out of the hug...
...in the dark and now the moon rises , full and amber a...
...bird on the shape of the moon over the trees in front...

A Naïve Approach


- Build a **co-occurrence matrix** for words, and apply SVD

- **Example Corpus:**

- I like deep learning.
- I like NLP.
- I enjoy flying.


- **Issues:**

- Global context
- SVD is very expensive



counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram 
- GloVe: Global Vectors for Word Representation
- Summary

Word2Vec

- Proposed by Mikolov et al. at Google in 2013
 - The most popular word embedding models
 - Won Test of Time Award of NeurIPS 2023



Jeff Dean (@JeffDean)

On behalf of our co-authors Tomáš Mikolov, @ilyasut and Kai Chen, @greg_corrado and I were delighted to accept the #NeurIPS2023 Test of Time Award for the "word2vec" paper (arxiv.org/abs/1310.4546 ✓). Thanks to the @NeurIPSConf test of time committee for honoring us with this award!

... We are honored to receive this award

- Thanks to the committee that selected our work! We're honored!
- Thanks to our co-authors who couldn't be here today!

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov



Ilya Sutskever



Kai Chen



Greg Corrado



Jeffrey Dean



Jeff Dean Facts

1. During his own Google interview, Jeff Dean was asked the implications if $P=NP$ were true. He said, " $P = 0$ or $N = 1$ ". Then, before the interviewer had even finished laughing, Jeff examined Google's public certificate and wrote the private key on the whiteboard.
2. Compilers don't warn Jeff Dean. Jeff Dean warns compilers.
3. The rate at which Jeff Dean produces code jumped by a factor of 40 in late 2000 when he upgraded his keyboard to USB 2.0.

"Jeff pumped out elegant code like a champagne fountain at a wedding. It seemed to pour from him effortlessly in endless streams that flowed together to form sparkling programs that did remarkable things. He once wrote a two-hundred-thousand-line application to help the Centers for Disease Control manage specialized statistics for epidemiologists. It's still in use and garners more peer citations than any of the dozens of patented programs he has produced in a decade at Google. He wrote it as a summer intern in high school."

<https://www.quora.com/Jeff-Dean/What-are-all-the-Jeff-Dean-facts>

Interviewer:

Jeff, is it true that Google Search was your Noogler Project, that you were unsatisfied with constant time and created the world's first $O(1/n)$ algorithm, and all the other 'facts' about you on <https://www.quora.com/What-are-all-the-Jeff-Dean-facts> ?

Jeff:

11111111

Interviewer:

That is not even English, what does that mean?

Jeff Dean:



Jeff Dean

EVERY BIT OF IT IS TRUE.

Ilya Sutskever

Co-founder and Chief Scientist of [OpenAI](#).

I spent three wonderful years as a Research Scientist at the Google Brain Team.

Before that, I was a co-founder of [DNNresearch](#).

And before that, I was a postdoc in Stanford with [Andrew Ng](#)'s group.

And in the beginning, I was a student in the Machine Learning group of Toronto, working with [Geoffrey Hinton](#).



<https://www.cs.toronto.edu/~ilya/>



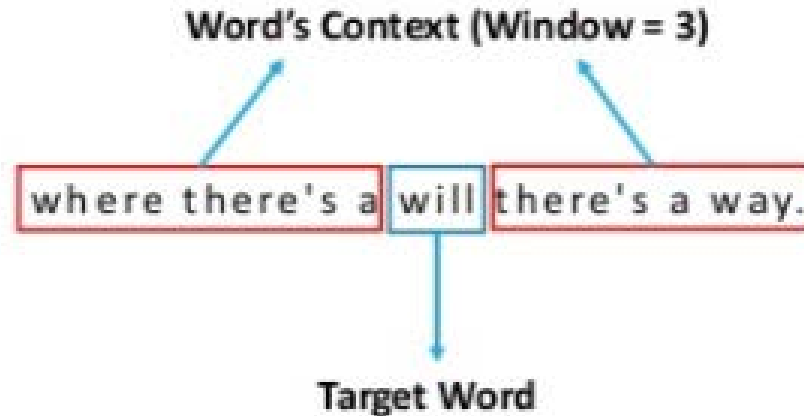
<https://mashable.com/article/openai-ilya-sutskever-deeply-regrets-sam-altman-firing>

Back to Word2Vec

- Two architectures are proposed
 - Continuous bag-of-words (CBOW)
 - Skip-gram
- Extremely fast
 - “an optimized single-machine implementation can train on more than 100 billion words in one day”

Main Idea of Word2Vec

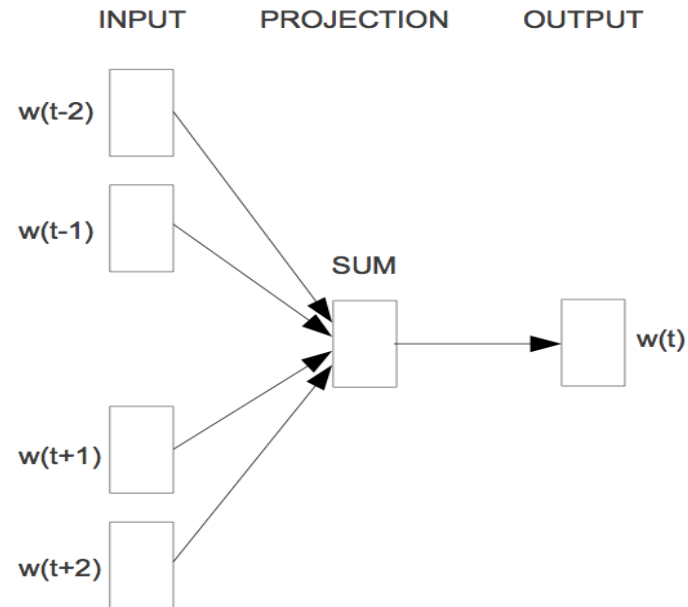
- Consider a local window of a target word



- **CBOW**: predict the target words given the neighbors
- **Skip-gram**: predict neighbors given the target words

CBOW

- Predicting target using neighbors



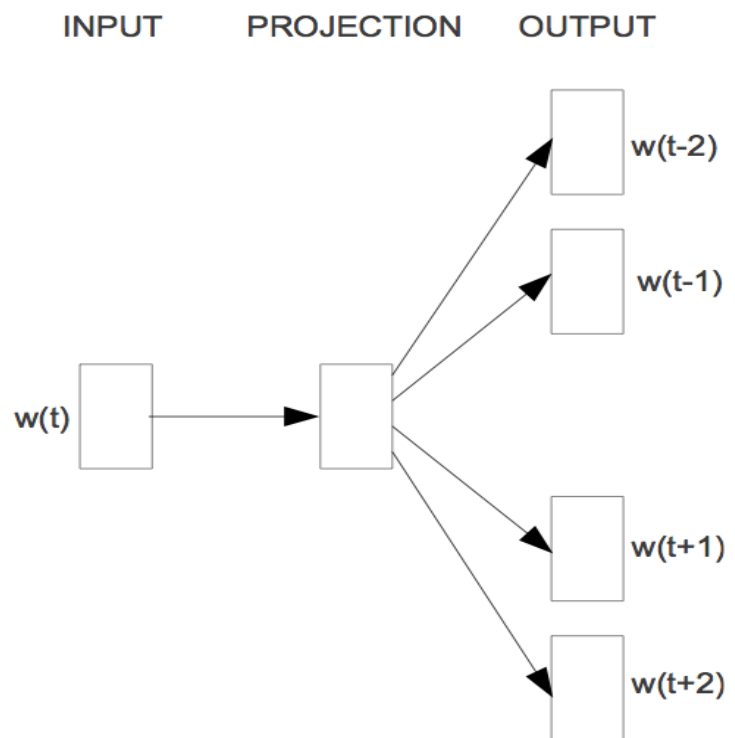
$$J_{\theta} = \frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n})$$

More details can be found in: <https://arxiv.org/pdf/1301.3781.pdf>

<https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>

Skip-Gram

- Predicting neighbors using target



$$J_{\theta} = \frac{1}{T} \sum_{t=1}^T \sum_{-n \leq j \leq n, j \neq 0} \log p(w_{t+j} | w_t)$$

The Conditional Probability

- $p(w_{t+j}|w_t)$: the probability to see w_{t+j} in target word w_t 's neighborhood
 - Intuition: w_t 's embedding should be closer to w_{t+j} 's embedding
 - Every word has two embedding vectors
 - One serves as the role of target ($\mathbf{v} \in R^{d \times 1}$), and the other serves as the role of context ($\mathbf{u} \in R^{d \times 1}$)

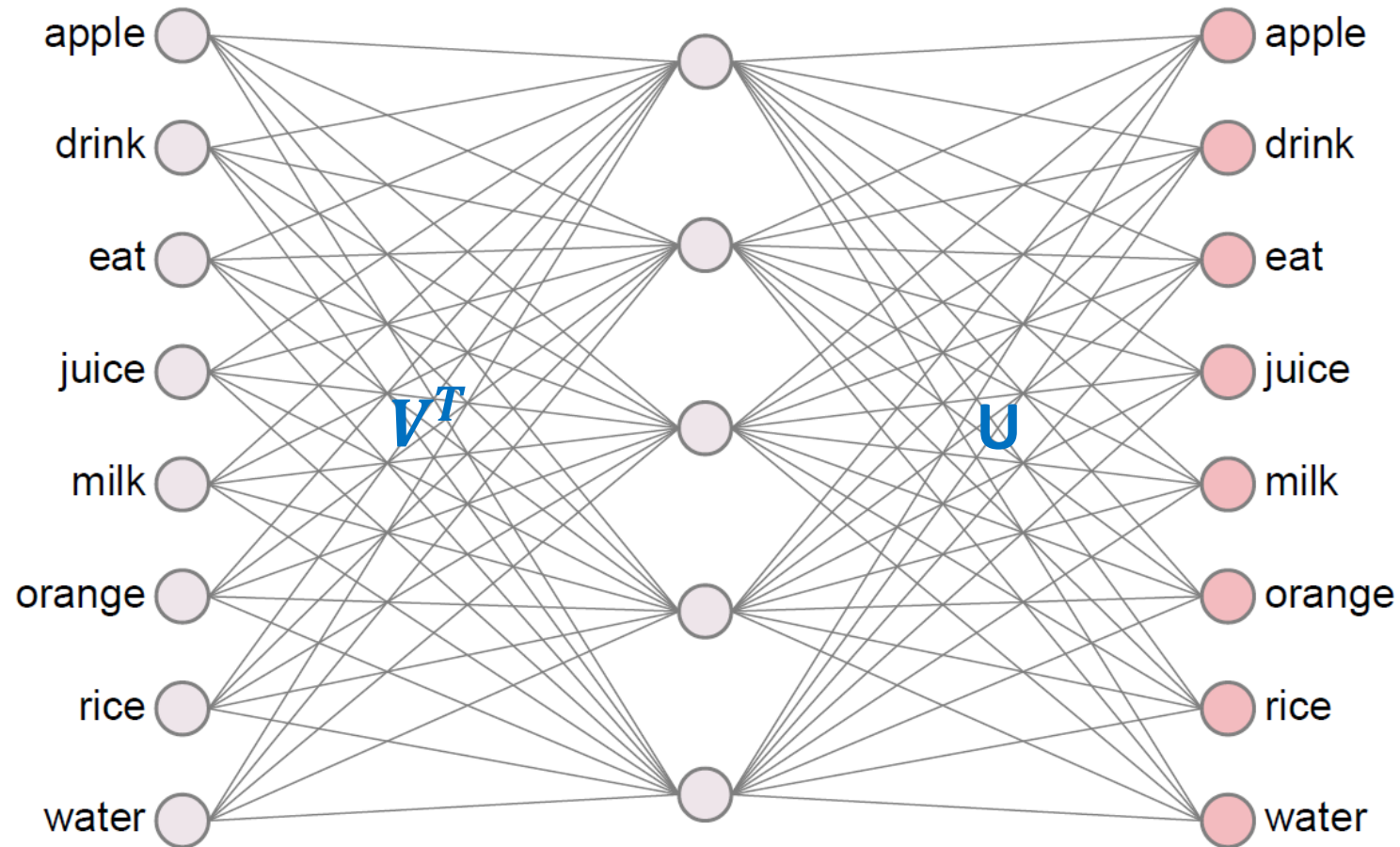
$$p(o|w) = \frac{\exp(u_o^T v_w)}{\sum_{w'=1}^N \exp(u_{w'}^T v_w)}$$

(N is the total number of words in vocabulary)

Transformation Function under the Neural Network

- Input: \mathbf{x}
 - One hot encoding vector
- Neural function: $\mathbf{y} = f(\mathbf{x})$
 - $f(\mathbf{x}) = \text{softmax}(UV^T \mathbf{x})$
 - $V: n \times d$
 - $U: n \times d$
- Output: \mathbf{y}
 - Probability vector indicating the probability to have each word in the vocabulary

A Neural Network Point of View



Input Layer:
one-hot vector

Hidden Layer:
Linear (Identity)

Output Layer:
softmax

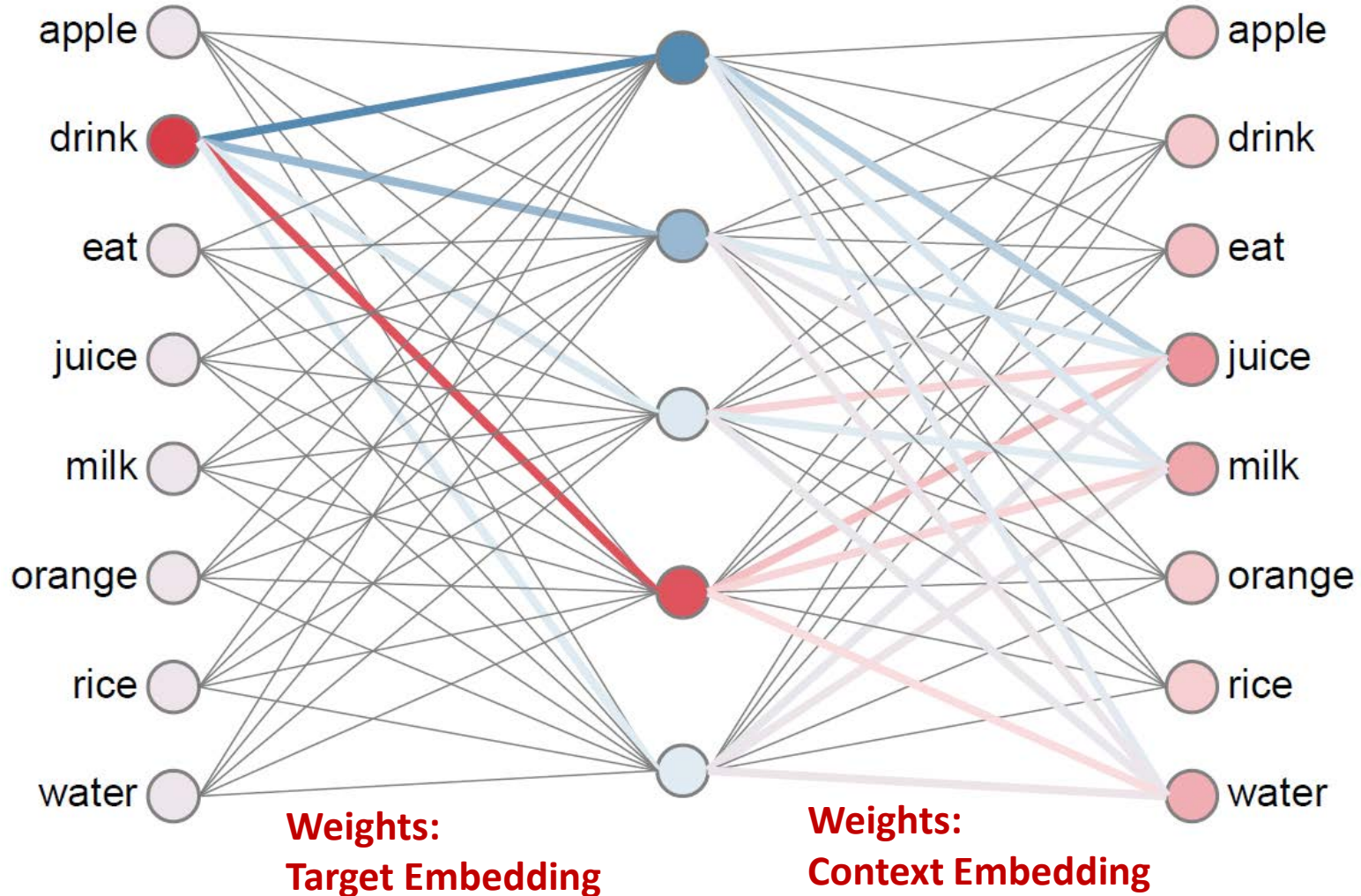
Question

- What's the dimension of input layer?
- What's the dimension of hidden layer?
- What's the dimension of output layer?

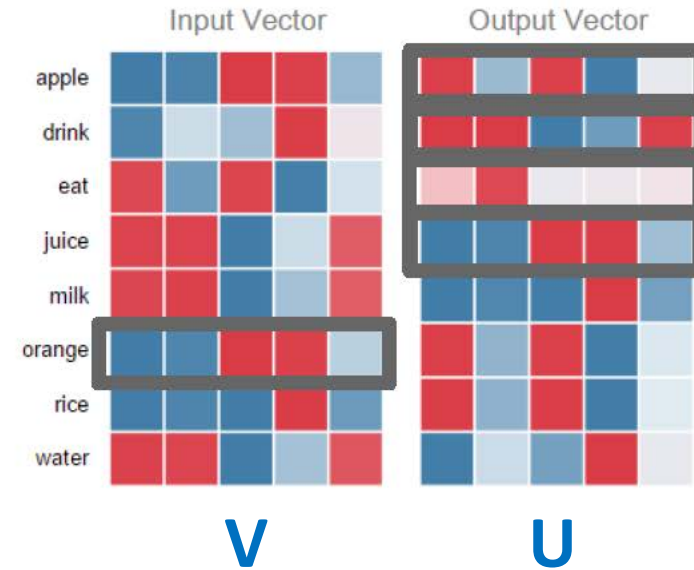
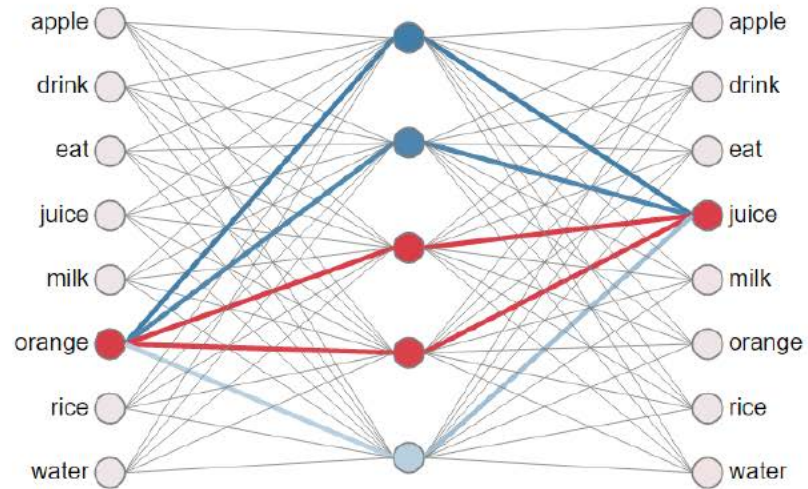
$$V: n \times d; U: n \times d$$

Demo

- <https://ronxin.github.io/wevi/>

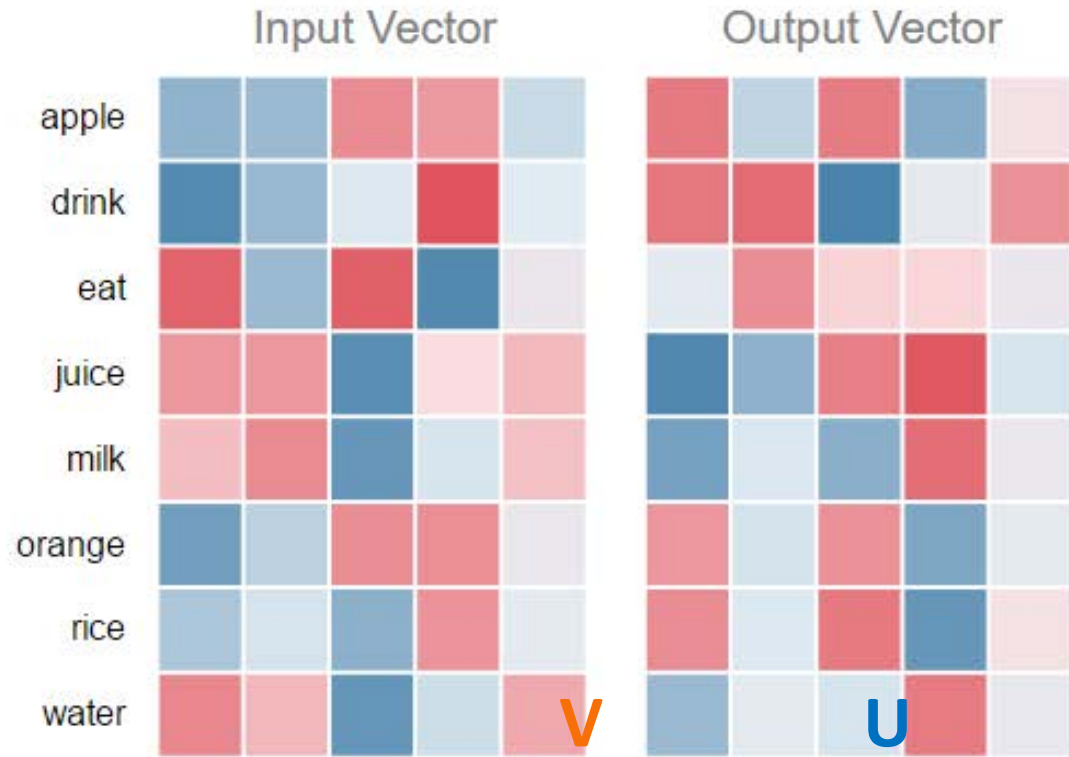


Embedding vs. NN Weights

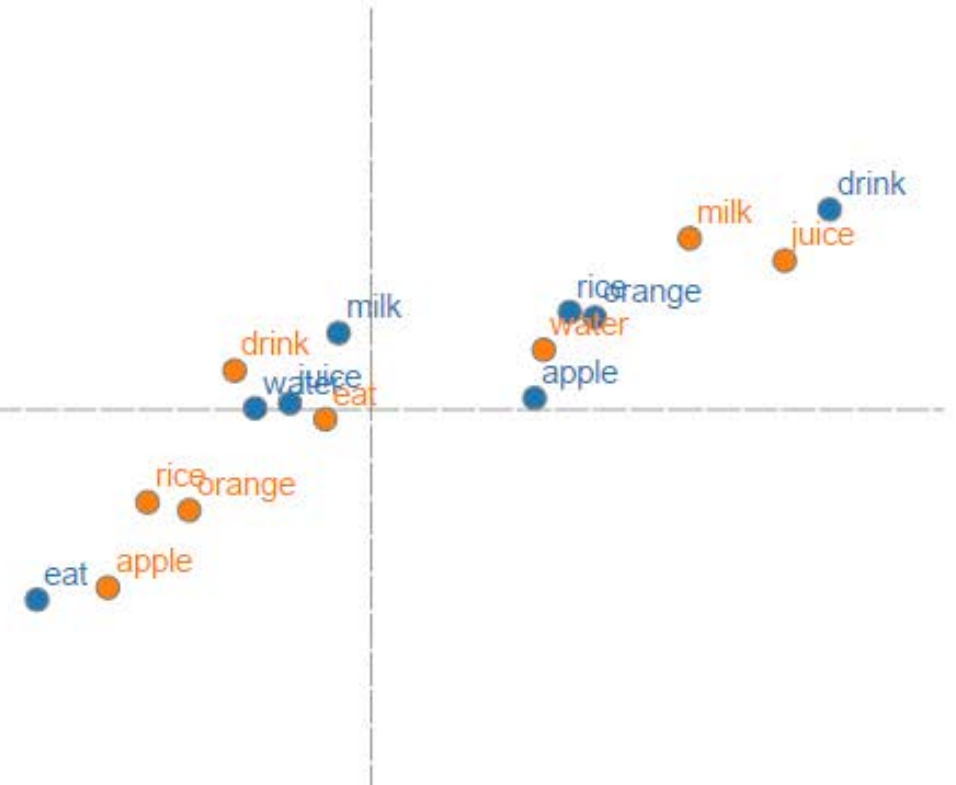


Embedding Visualization

Weight Matrices



Vectors



Original Objective Function

- The original objective is not scalable for large size vocabulary!

$$p(o|w) = \frac{\exp(u_o^T v_w)}{\sum_{w'=1}^N \exp(u_{w'}^T v_w)}$$

- Maximize: $\prod_{(w,c) \in D} p(c|w)$
 - (w, c) denote any target word and context word pair

Negative Sampling for Skip-Gram

- For each target, for every positive word, sample k negative words

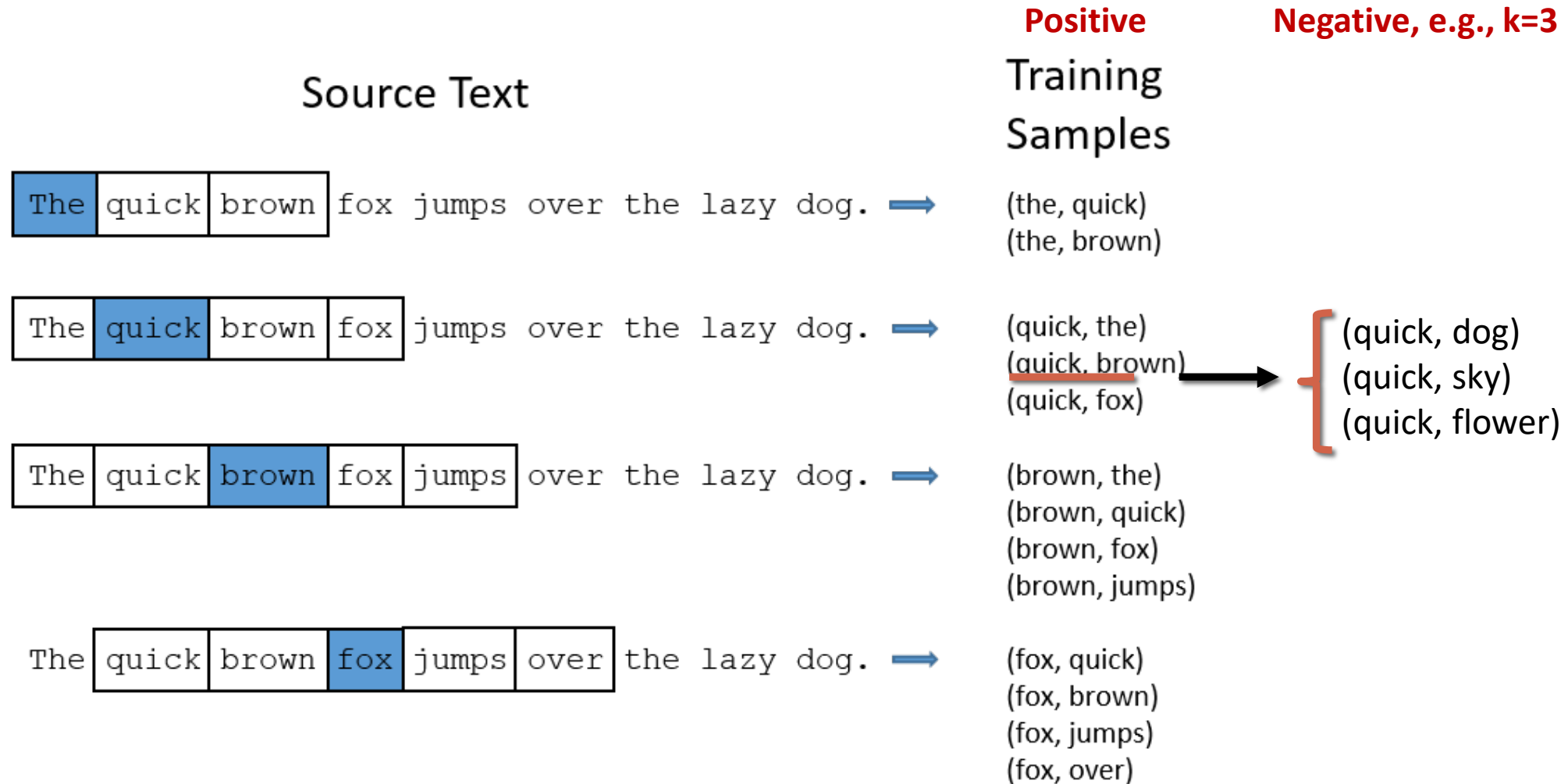
$$\sum_{(w,c) \in D} \left[\log \sigma(u_c^T v_w) + \sum_{i=1}^k E_{w_i \sim P_n(w)} [\log \sigma(-u_{w_i}^T v_w)] \right]$$

$\sigma(\cdot)$: sigmoid function

$P_n(w)$: “Negative” Distribution

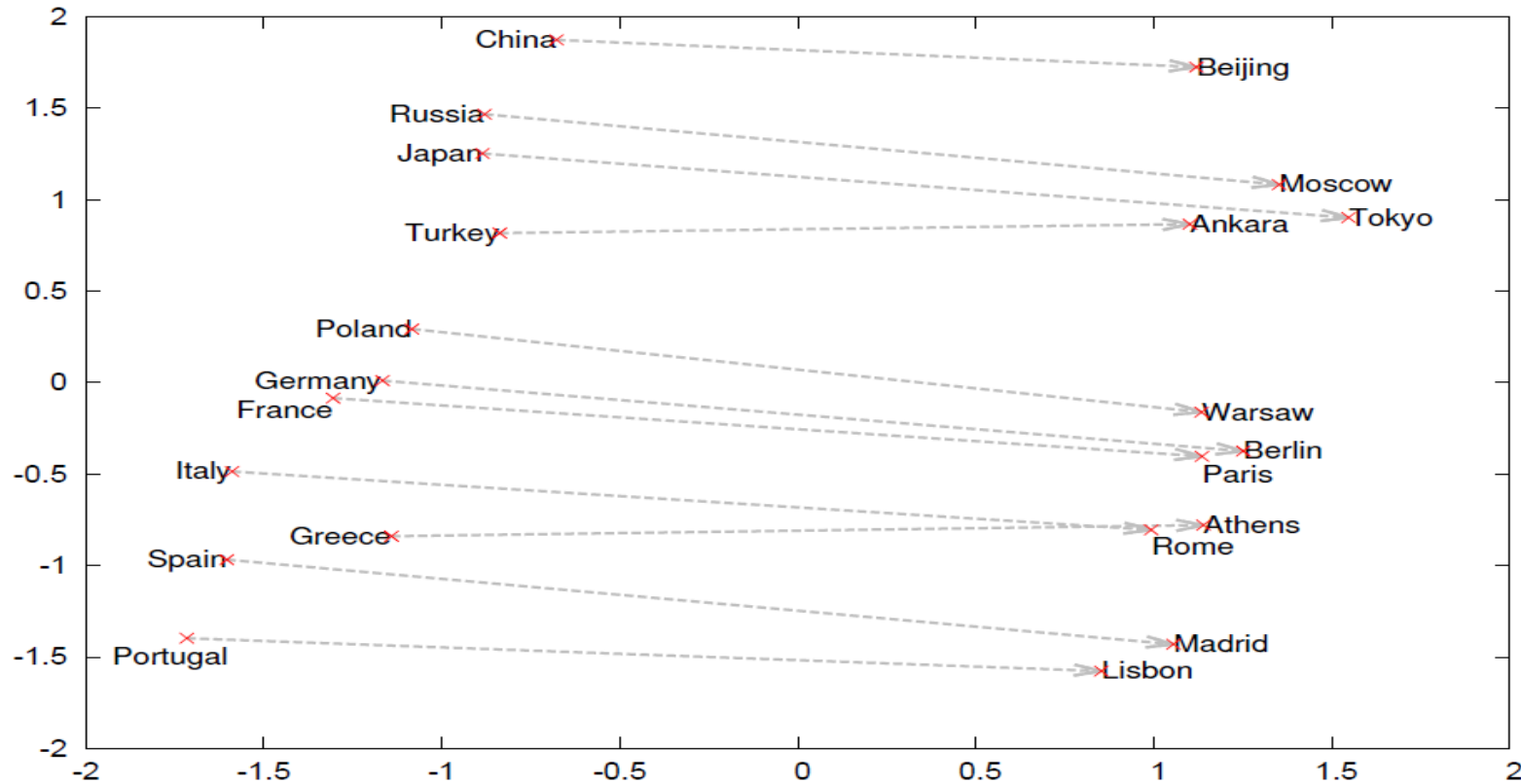
- Examples: (1) $\propto \text{count}(w)$; (2) $\propto (\text{count}(w))^{\frac{3}{4}}$, where $\text{count}(w)$ is the total count of w

More on Negative Samples




A Potential Application

- Relation detection and knowledge completion



$$v_{Berlin} - v_{Germany} \approx v_{Paris} - v_{France}$$

Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation 
- Summary

Combining Two Worlds

- Matrix factorization for global word-word co-occurrence matrix
 - E.g., SVD
 - Global matrix factorization
- Make predictions within local context windows
 - E.g., word2vec
 - Local context window

Objective Function

$$J = \sum_{i,j=1}^V f(X_{ij}) \left(\boxed{w_i^T \tilde{w}_j + b_i + \tilde{b}_j} - \boxed{\log X_{ij}} \right)^2$$

Predicted value observed value

X_{ij} : number of times word j appears in the context of word i

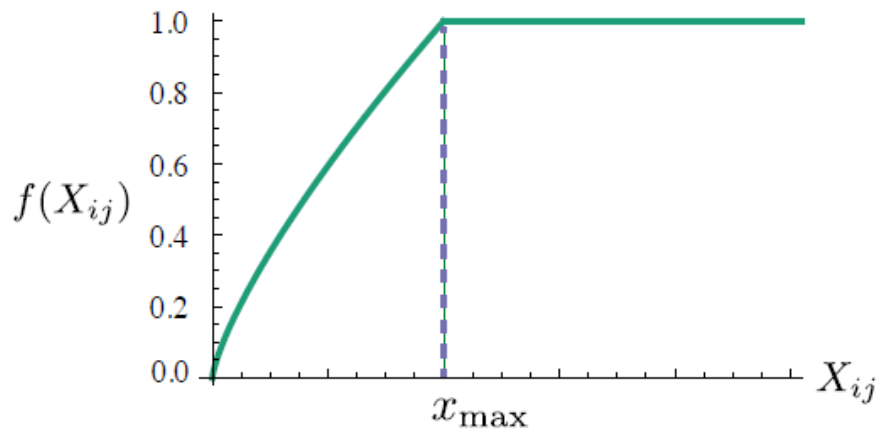
w_i : word vector for word i

\tilde{w}_j : context word vector for word j

b_i : bias term for word i

\tilde{b}_j : bias term for context word j

$f(X_{ij})$: a weighting function to punish rare words



$$f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

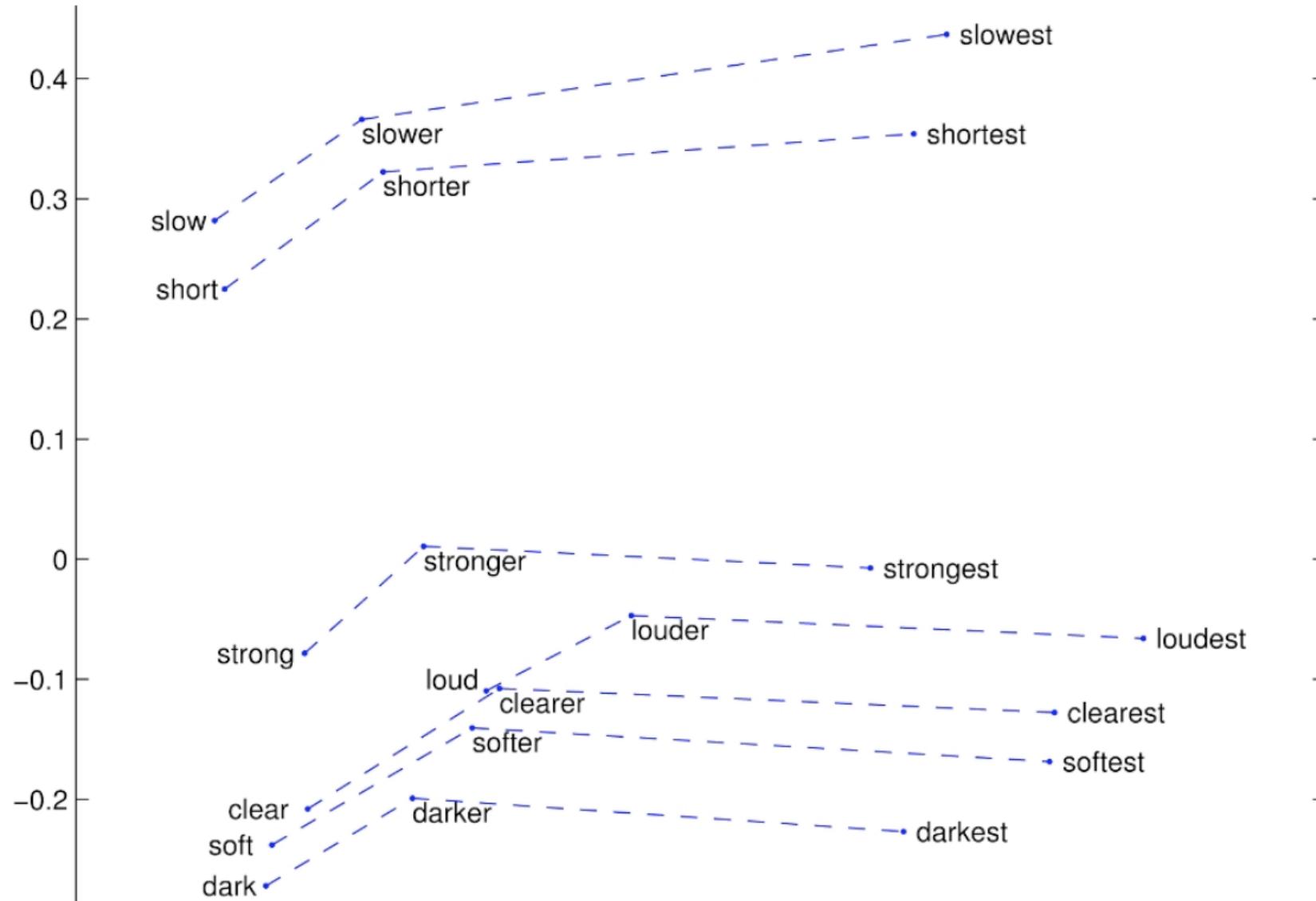
<https://nlp.stanford.edu/pubs/glove.pdf>

Weighting function f with $\alpha = 3/4$

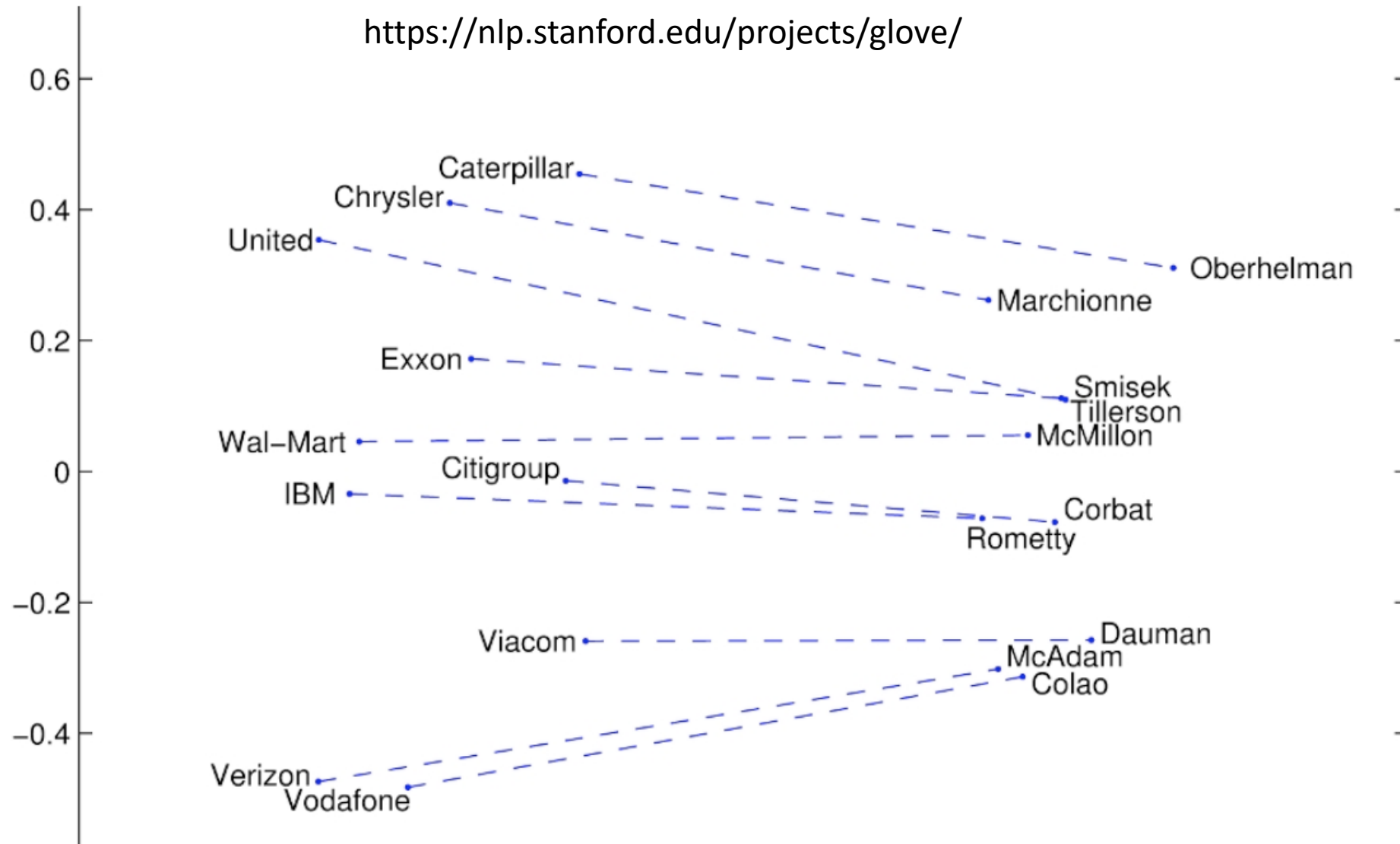
Question

- How to optimize the objective function?


Some Interesting Results: Superlatives



Some Interesting Results: Company-CEO



Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary 

Summary

- Word embedding
 - A low-dimensional vector representation for words
- Word2vec
 - Local context-based prediction: CBOW and Skip-Gram
- Glove
 - Matrix decomposition on local context co-occurrence matrix

References

- Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 1–12.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS, 1–9.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532–1543.
- Yoav Goldberg and Omer Levy (2014). Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method.
<https://arxiv.org/pdf/1402.3722v1.pdf>